UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/580,986	05/25/2006	Bo Huang	42P22899	5991

45209 7590 11/01/2010
INTEL/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
MITCHELL, JASON D	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
11/01/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/580,986	HUANG ET AL.
	Examiner	Art Unit
	JASON MITCHELL	2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 August 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

This action is in response to an amendment filed on 8/19/10.

Claims 1-24 are pending in this application.

Response to Arguments

Objection to Claim 1

The amendment to claim 1 is sufficient to overcome the previous objection which is consequently withdrawn.

Rejection of Claims Under 35 USC 101

The amendments to claims 1 and 21 are sufficient to overcome the previous 35 USC 101 rejections which have consequently been withdrawn.

Rejection of Claims Under 35 USC 102

The applicants' amendment required new grounds of rejection. Specifically the Chow reference has been replaced with US 2001/0032332 to Ward et al (Ward). Accordingly arguments regarding the Chow reference are now moot. To the extent that the applicants arguments are still relevant they have been found unpersuasive as will be discussed below.

In the par. bridging pp. 10 and 11, the applicants state:

> Lagergren discloses that the size of the application code is used as the size metric for performing optimization. See Lagergren paragraph [0016]. The size metric is calculated using factors, together with associated weights. Id., paragraph [0025]. These factors may include basic block count, operation tuple count, register pressure, recognized patterns, preference for aggressive in-lining, maximum population count of a basic block live inset, control flow, and the number of code operations. Id., paragraphs [0025], [0026], and [0028]. In other words, Lagergren teaches using code size to compute a figure of merit (i.e., size metric), and the factors are only used to inform this sole figure of merit. Thus, Lagergren arguably teaches *a second evaluator to measure code size and compute a code size figure of merit based on the measured code size*, as recited in Applicants' claim 1. The method of calculating the size metric in Lagergren could arguably be used to evaluate another characteristic such as those disclosed in Chow. However, it is not obvious to one of ordinary skill in the art to go one step further and use the method taught in Lagergren to *compute an overall figure of merit as a function of the power consumption figure of merit based on power consumption and the code size figure of merit based on code size*, as recited in Applicants' claim 1.

The examiner respectfully disagrees. While Lagergren's overall metric is disclosed as a 'size' metric, so are, at least many of, the 'weighted factors' used to calculate the overall size metric (e.g. basic block count, operation tuple count, number of code operations). Accordingly, Lagergren teaches an overall size metric (i.e. dynamic size) calculated a function of component size metrics (e.g. basic block count, operation tuple count, number of code operations). This parallels, and thus teaches, computing an overall optimization metric (i.e. "an overall figure of merit") as a function of component optimization metrics (e.g. "power consumption" and "code size").

As will be discussed further in the rejection, Ward discloses a system which collects and compares power consumption, size and performance metrics (see e.g. par. [0020] "performance metrics … such as size, power consumed, and cycles consumed"). This, in conjunction with the teachings of Lagergren, makes obvious the claimed computing an overall figure of merit.

Claim Objections

Claim 6 is objected to because of the following informalities:

Claim 6 recites “the respective power consumption figures of merit and code size figure of merit of that input binary”. It is believed this should read “the respective power consumption figures of merit and code size figures of merit of that input binary” or alternately “the respective power consumption figure[[s]] of merit and code size figure of merit of that input binary”

Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-7, 13-15, 17, 19, 21 and 23-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 2001/0032332 to Ward et al. (Ward) in view of US 2004/0117779 to Lagergren (Lagergren).

Claim 1: Ward discloses a system for evaluating and selecting programming code, comprising:

a first evaluator, executed by a computer system, to measure power consumption of a plurality of input binaries and compute a plurality of power consumption figures of merit for the plurality of input binaries, respectively, based on the measured power consumption (par. [0020] “measured performance metrics for each function in the program (such as … power consumed …)”);

a second evaluator, executed by the computer system, to measure code size of the plurality of input binaries and compute a plurality of code size figures of merit for the plurality of input binaries, respectively, based on the measured code size (par. [0020] “measured performance metrics for each function in the program (such as size …)”); and

comparing the plurality of figures of merit with each other to select one of the plurality of input binaries as having the most merit (par. [0039] “allows the user using the user interface 17 to select a solution point”; also see Fig. 6).

Ward does not disclose a binary selector, executed by the computer system, to compute a plurality of overall figures of merit for the plurality of input binaries and to compare the plurality of overall figures of merit with each other to select one of the plurality of input binaries as having the highest or lowest overall figure of merit.

Lagergren teaches a binary selector, executed by the computer system, to compute a plurality of overall figures of merit for a plurality of input binaries (par. [0025] "The output of the code introspection process is a number of factors, together with associated

weights that are then used by the system in calculating the size metric”), respectively, wherein each overall figure of merit (par. [0025] "the size metric”) is computed as a function of respective individual figures of merit (par. [0025] "a number of factors, together with associated weights”),

the binary selector to compare the plurality of overall figures of merit with each other to select one of the plurality of input binaries as having the highest or lowest overall figure of merit (par. [0025] "factors, together with associated weights that are then used … in calculating the size metric”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to compute an overall figure of merit (Lagergren par. [0025] "factors, together with associated weights that are then used … in calculating the size metric”) by using Ward’s individual figures of merit (par. [0020] “measured performance metrics for each function in the program (such as size, power consumed …)”). Those of ordinary skill in the art would have been motivated to do so as a means of automating, at least to an extent, the selection process disclosed in Ward as being performed by a user (par. [0039] “allows the user using the user interface 17 to select a solution point”). In other words, Ward discloses displaying to a user a graph indicating the figures of merit measured for each of the plurality of binaries (e.g. Ward Fig. 6) and allowing the user to select a binary which provides a desired overall optimization (Ward par. [0039] “allows the user using the user interface 17 to select a solution point”). By performing a calculation similar to that taught by Lagergren, Ward's two dimensional graph could be

converted into a linear representation and the ‘best' optimization could be chosen by the computer (par. [0025] "factors, together with associated weights that are then used … in calculating the size metric”).

Claim 2: The rejection of claim 1 is incorporated; further Ward discloses the binary selector is to compare the plurality of power consumption figures of merit with each other and to compare the plurality of second code size figures of merit with each other (par. [0032] “compute all of the useful solutions. ... A solution is "useful", if no other solution results in a version of the application that is better optimized for a specific result, such as., for example, faster and small in code size than it").

Claim 3: The rejection of claim 1 is incorporated further Lagergren teaches the binary selector is to compute the plurality of overall figures of merit by assigning a first weight to the plurality of power consumption figures of merit and a second weight to the plurality of code size figures of merit (par. [0025] "factors, together with associated weights).

Claim 4: The rejection of claim 1 is incorporated; further Ward and Lagergren teach:

a third evaluator, executed by the computer system, to measure performance of the plurality of input binaries and compute a plurality of performance figures of merit for the plurality of input binaries, respectively (par. [0020] “measured performance metrics for each function in the program (such as …cycles consumed.)”), wherein

the binary selector is to compare the plurality of performance figures of merit with each other (Ward Fig. 6; Lagergren par. [0025] "factors, together with associated weights that are then used … in calculating the size metric”).

Claim 5: The rejection of claim 4 is incorporated; further Ward discloses the greater the performance the smaller its associated figure of merit, and the smaller the code size the smaller its associated figure of merit (par. [0035] "records will be ... clearly inferior (their time and size are both greater than the others)").

Claim 6: The rejection of claim 2 is incorporated; further Ward and Lagergren teach the binary selector is to compare the plurality of power consumption figures of merit and code size figures of merit by computing a mathematical operation for each of the input binaries which includes the respective power consumption figures of merit and code size figure of merit of that input binary (Ward par. [0020] “measured performance metrics for each function in the program (such as size, power consumed, and cycles consumed.”; Lagergren par. [0025] "The output of the code introspection process is a number of factors, together with associated weights that are then used by the system in calculating the size metric”).

Claim 7: The rejection of claim 1 is incorporated; further Ward discloses wherein the code size includes at least one of code size (par. [0020] “such as size”), compressed file size, and memory footprint.

Claim 13: Ward discloses a machine-implemented method for processing computer programming code, comprising:

a) producing a current version of a binary using a current optimization setting (par. [0019] “compiled … with different compiling options”);

b) measuring code size of the current version and computing a current code size figure of merit (FOM) associated with the current version based on the measured code size (par. [0020] “measured performance metrics for each function in the program (such as size …)”);

c) measuring performance of the current version and computing a current performance FOM associated with the current version based on the measured performance (par. [0020] “measured performance metrics for each function in the program (such as …cycles consumed.)”);

d) comparing the comparing the current FOM with a previously computed overall FOM associated with a prior version of the binary (par. [0039] “allows the user using the user interface 17 to select a solution point”; also see Fig. 6); and

automatically repeating a)-d) for another optimization setting (par. [0019] “compiled … with different compiling options”).

Further, it is noted that the claim does not appear to require that the "automatically repeating a)-d) for another optimization setting” be performed only after steps "a)-d)" are completed for the "current optimization setting". Accordingly, although Ward appears to describe, for example, performing step “a)” for each optimization

setting before moving on to step “b)”, it is not felt this constitutes a patentable distinction from the claims. But, regardless, performing the steps described in Ward in a more sequential manner would appear to fall within the “Various modifications and combinations” which would have been obvious to those of ordinary skill in the art (see par. [0058]).

Ward does not disclose computing a current overall FOM as a function of the current code size FOM and the current performance FOM and using the overall FOM as a point of comparison.

Lagergren teaches computing a current overall FOM (par. [0025] "the size metric”) as a function of individual FOMs (par. [0025] "a number of factors, together with associated weights”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to compute an overall FOM (Lagergren par. [0025] "factors, together with associated weights that are then used … in calculating the size metric”) by using Ward’s individual FOMs (par. [0020] “measured performance metrics for each function in the program (such as size, power consumed …)”). Those of ordinary skill in the art would have been motivated to do so as a means of automating, at least to an extent, the selection process disclosed in Ward as being performed by a user (par. [0039] “allows the user using the user interface 17 to select a solution point”). In other words, Ward

discloses displaying to a user a graph indicating the figures of merit measured for each of the plurality of binaries (e.g. Ward Fig. 6) and allowing the user to select a binary which provides a desired overall optimization (Ward par. [0039] “allows the user using the user interface 17 to select a solution point”). By performing a calculation similar to that taught by Lagergren, Ward's two dimensional graph could be converted into a linear representation and the ‘best' optimization could be chosen by the computer (par. [0025] "factors, together with associated weights that are then used … in calculating the size metric”).

Claim 14: The rejection of claim 13 is incorporated; further Ward and Lagergren teach:

indicating to a user the version of the binary that has the highest or lowest overall FOM as determined from the comparisons (Ward Fig. 6; Lagergren Fig. 2, Step 30).

Claim 15: The rejection of claim 14 is incorporated; further Ward and Lagergren teach:

ranking a plurality of versions of the binary in accordance with their respective overall FOMS as determined from the comparisons (Ward Fig. 6; Lagergren Fig. 2, Step 30, par. [0025] "The output of the code introspection process is a number of factors, together with associated weights that are then used by the system in calculating the size metric”).

Claim 17: The rejection of claim 13 is incorporated; further Ward discloses wherein the current and another optimization settings include optimization controls for compilation and linking, and wherein said producing comprises:

compiling source code and linking object files to produce the current version of the binary, using the current optimization setting (par. [0019] “the whole program is compiled … in the compiler 13 … The result of the compiling step is a set of executable objects”; it should be understood that the referenced compiler includes a linker, otherwise the compilation would not result in an executable).

Claim 19: The rejection of claim 13 is incorporated; further Ward discloses:

e) comparing the current code size FOM with a previously computed code size FOM that is associated with code size and with the prior version of the binary (par. [0032] “compute all of the useful solutions. ... A solution is "useful", if no other solution results in a version of the application that is better optimized for a specific result, such as., for example, … smaller in code size than it"); and

f) comparing the current performance FOM with a previously computed performance FOM that is associated with the performance and with the prior version of the binary (par. [0032] “compute all of the useful solutions. ... A solution is "useful", if no other solution results in a version of the application that is better optimized for a specific result, such as, for example, faster ").

Claim 21: Ward discloses an article of manufacture comprising:

a non-transitory machine-accessible storage medium containing instructions that, when executed, cause a machine to:

a) generate a binary under an optimization setting (par. [0019] “compiled … with different compiling options”);

b) compute a performance cost as a function of a measured performance of the binary (par. [0020] “measured performance metrics for each function in the program (such as …cycles consumed.)”);

c) compute a power consumption cost as a function of a measured power consumption of the binary (par. [0020] “measured performance metrics for each function in the program (such as … power consumed …)”),

e) perform a)-d) a plurality of times each time with a different optimization setting but based on the same source program (par. [0019] “compiled … with different compiling options”); and

f) compare the costs with each other, to select the binary having the lowest cost (par. [0039] “allows the user using the user interface 17 to select a solution point”; also see Fig. 6).

Ward does not disclose computing an overall cost for the binary as a function of the performance cost and the power consumption cost.

Lagergren teaches computing an overall cost (par. [0025] "the size metric") for a binary as a function of individual measured costs (par. [0025] "a number of factors, together with associated weights")

It would have been obvious to one of ordinary skill in the art at the time the invention was made to compute an overall cost (Lagergren par. [0025] "factors, together with associated weights that are then used … in calculating the size metric") by using Ward's individual measured costs (par. [0020] "measured performance metrics for each function in the program (such as size, power consumed …)"). Those of ordinary skill in the art would have been motivated to do so as a means of automating, at least to an extent, the selection process disclosed in Ward as being performed by a user (par. [0039] "allows the user using the user interface 17 to select a solution point"). In other words, Ward discloses displaying to a user a graph indicating the figures of merit measured for each of the plurality of binaries (e.g. Ward Fig. 6) and allowing the user to select a binary which provides a desired overall optimization (Ward par. [0039] "allows the user using the user interface 17 to select a solution point"). By performing a calculation similar to that taught by Lagergren, Ward's two dimensional graph could be converted into a linear representation and the 'best' optimization could be chosen by the computer (par. [0025] "factors, together with associated weights that are then used … in calculating the size metric").

Claim 23: The rejection of claim 21 is incorporated; further Ward discloses instructions that cause the machine to perform c) by computing a code size cost as a function of a measured code size of the binary generated in a) (par. [0020] “measured performance metrics for each function in the program (such as size …)”).

Claim 24: The rejection of claim 23 is incorporated; further, Ward and Lagergren teach the instructions cause the machine to compare the computed overall costs in f), by computing an overall cost for each generated binary, wherein the overall cost is a function of the performance cost, the power consumption cost, and the code size cost (Ward par. [0020] “such as size, power consumed, and cycles consumed”; Lagergren par. [0025] "The output of the code introspection process is a number of factors, together with associated weights that are then used by the system in calculating the size metric”).

Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 2001/0032332 to Ward et al. (Ward) in view of US 2004/0117779 to Lagergren (Lagergren) in view of “Feedback-Directed Selection and Characterization of Compiler Optimizations” by Chow and Wu (Chow).

Claim 8: The rejection of claim 7 is incorporated; further Ward discloses:

a code generator, executed by the computer system, that includes a compiler and a linker to process an output of the compiler and produce the input binaries (par.

[0019] “the whole program is compiled … in the compiler 13 … The result of the compiling step is a set of executable objects”; it should be understood that the referenced compiler includes a linker, otherwise the compilation would not result in an executable), wherein

the compiler, executed by the computer system, exposes an optimization control to its user (par. [0019] “compiled … with different compiling options”).

Ward and Lagergren do not explicitly teach the optimization control is selected from the group consisting of: loop-unrolling; vectorization; and constant propagation.

Chow teaches a compiler optimization control selected from the group consisting of: loop-unrolling (pg. 2, col.1, 2nd par. "loop-unrolling transformation"); vectorization; and constant propagation.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to expose an optimization control to the user (Ward par. [0019] “compiled … with different compiling options”) selected from the group consisting of: loop-unrolling (Chow pg. 2, col.1, 2nd par. "loop-unrolling transformation"); vectorization; and constant propagation. Those of ordinary skill in the art would have been motivated to do so because Ward discloses the use of optimization controls generally and those of ordinary skill in the art would have implemented the optimizations using known techniques (e.g. Chow pg. 2, col.1, 2nd par. "loop-unrolling transformation").

Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 2001/0032332 to Ward et al. (Ward) in view of US 2004/0117779 to Lagergren (Lagergren) in view of "Feedback-Directed Selection and Characterization of Compiler Optimizations" by Chow and Wu (Chow) in view of Applicant Acknowledged Prior Art Techniques (AAPA).

Claim 9: The rejection of claim 8 is incorporated; further Ward, Lagergren and Chow do not disclose the code generator further comprises a binary rewriter to process an output of the linker and produce the input binaries, wherein the binary rewriter exposes an optimization control to its user selected from the group consisting of: constant propagation; code shrinking; and specialization.

The applicants acknowledge that binary rewriters exposing an optimization control to its user selected from the group consisting of constant propagation; code shrinking; and specialization were known in the prior art (par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a binary rewriter in the generation of an executable to expose an optimization control to the user (Ward par. [0019] "compiled … with different compiling

options") selected from the group consisting of: constant propagation; code shrinking; and specialization were known in the prior art (AAPA par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization"). Those of ordinary skill in the art would have been motivated to do so because Ward discloses the use of optimization controls generally and those of ordinary skill in the art would have implemented the optimizations using known techniques (e.g. AAPA par. bridging pp. 9-10 " Example optimization controls include constant propagation; code shrinking and specialization").

Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 2001/0032332 to Ward et al. (Ward) in view of US 2004/0117779 to Lagergren (Lagergren) in view of "Feedback-Directed Selection and Characterization of Compiler Optimizations" by Chow and Wu (Chow) in view of US 2006/0064676 to Chavan (Chavan).

Claim 10: The rejection of claim 8 is incorporated; further Ward discloses:

Defining a plurality of optimization combinations and configuring the code generator in accordance with the optimization combinations (par. [0019] "the whole program is compiled several times in the compiler 13, each time with different compiling option"), wherein

the code generator is to produce the input binaries as configured by the optimization combinations, respectively (par. [0019] “The result of the compiling step is a set of executable objects, one for each option set”).

Ward, Lagergren and Chow do not teach a script process to process an input script from the user defining the plurality of optimization combinations.

Chavan teaches a script processor to process an input script containing a plurality of optimization combination to configure a code generator in accordance with the optimization combinations (par. [0043] “script 432 can ... invoke compiler 414 with ... compiler optimization options enabled … to generate optimized object code 422”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use a script processed by a script processor (Chavan par. [0043] “script 432 can ... invoke compiler 414 with ... compiler optimization options enabled … to generate optimized object code 422”) to indicate the plurality of optimization combinations to the compiler (par. [0019] “The option sets are chosen at control 13a … to optimize the different performance metrics”). Those of ordinary skill in the art would have been motivated to do so in order to automate what is disclosed as a manual process (i.e. Ward par. [0019] “The option sets are chosen at control 13a by the user using a graphical user interface 17”).

Claims 11, 16, 18 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 2001/0032332 to Ward et al. (Ward) in view of US 2004/0117779 to Lagergren (Lagergren) in view of Applicant Acknowledged Prior Art Techniques (AAPA).

Claim 11: The rejection of claim 7 is incorporated; further Ward and Lagergren do not teach a binary rewriter, executed by the computer system, to produce the input binaries based on a source binary.

The applicants acknowledge that binary rewriters to produce input binaries based on a source binary, wherein the binary rewriters expose optimization controls to their users were known in the prior art (par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization")

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a binary rewriter in the generation of an executable to expose an optimization control to the user (Ward par. [0019] "compiled … with different compiling options"; AAPA par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization"). Those of ordinary skill in the art would have been motivated to do so because Ward discloses the use of optimization controls generally

and those of ordinary skill in the art would have implemented the optimizations using known techniques (e.g. AAPA par. bridging pp. 9-10 " Example optimization controls include constant propagation; code shrinking and specialization”).

Claim 16: The rejection of claim 13 is incorporated; further Ward discloses the current and another optimization settings include optimization controls for compilation, and linking, wherein said producing comprises:

compiling source code and linking object files to produce an initial version of the binary, using the current optimization setting (par. [0019] “the whole program is compiled … in the compiler 13 … The result of the compiling step is a set of executable objects”; it should be understood that the referenced compiler includes a linker, otherwise the compilation would not result in an executable).

Ward and Lagergren do not teach binary rewriting to produce binaries using optimization settings.

The applicants acknowledge that binary rewriting to produce binaries using optimization settings were known in the prior art (par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization”)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a binary rewriter in the generation of an executable to provide additional optimization settings (Ward par. [0019] “compiled … with different compiling options”; AAPA par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization”). Those of ordinary skill in the art would have been motivated to do so because Ward discloses the use of optimization controls generally and those of ordinary skill in the art would have implemented the optimizations using known techniques (e.g. AAPA par. bridging pp. 9-10 " Example optimization controls include constant propagation; code shrinking and specialization”).

Claim 18: The rejection of claim 13 is incorporated; further Ward and Lagergren do not teach the current and another optimization settings include optimization controls for binary rewriting.

The applicants acknowledge that binary rewriting to produce binaries using optimization settings were known in the prior art (par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization”)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a binary rewriter in the generation of an executable to provide

additional optimization settings (Ward par. [0019] “compiled … with different compiling options”; AAPA par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization”). Those of ordinary skill in the art would have been motivated to do so because Ward discloses the use of optimization controls generally and those of ordinary skill in the art would have implemented the optimizations using known techniques (e.g. AAPA par. bridging pp. 9-10 " Example optimization controls include constant propagation; code shrinking and specialization”).

Claim 22: The rejection of claim 21 is incorporated; further Ward discloses the instructions cause the machine to perform a)-d) a plurality of times, by first compiling the source program and then recompiling the source program (par. [0019] “The result of the compiling step is a set of executable objects, one for each option set”).

Ward and Lagergren do not teach rewriting the binary a plurality of times.

The applicants acknowledge that binary rewriting to rewrite binaries was known in the art (par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to rewrite the binaries to apply optimizations (Ward par. [0019] “compiled … with different compiling options”; AAPA par. bridging pp. 9-10 "The binary rewriter 304 may be a conventional, binary rewriting tool ... Example optimization controls include constant propagation; code shrinking and specialization”). Those of ordinary skill in the art would have been motivated to do so because Ward discloses the use of optimization controls generally and those of ordinary skill in the art would have implemented the optimizations using known techniques (e.g. AAPA par. bridging pp. 9-10 " Example optimization controls include constant propagation; code shrinking and specialization”).

Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 2001/0032332 to Ward et al. (Ward) in view of US 2004/0117779 to Lagergren (Lagergren) in view of Applicant Acknowledged Prior Art Techniques (AAPA) in view of US 2006/0064676 to Chavan (Chavan).

Claim 12: The rejection of claim 11 is incorporated further Ward, Lagergren and AAPA teach a plurality of optimization combinations (Ward par. [0019] “compiled … with different compiling options”) and configuring the binary rewriter in accordance with the optimization combinations (AAPA par. bridging pp. 9-10 "Example optimization controls include constant propagation; code shrinking and specialization”), wherein

the binary rewriter is to produce the input binaries as configured by the optimization combinations, respectively (AAPA par. bridging pp. 9-10 “binaries 106 are generated by a binary rewriter 304”).

Ward, Lagergren and AAPA do not teach a script processor, executed by the computer system, to process an input script from the user, the script processor to read a plurality of optimization combinations from the input script and configure the binary rewriter in accordance with the optimization combinations

Chavan teaches a script processor to process an input script containing a plurality of optimization combinations to configure a code generator in accordance with the optimization combinations (par. [0043] “script 432 can ... invoke compiler 414 with ... compiler optimization options enabled … to generate optimized object code 422”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use a script processed by a script processor (Chavan par. [0043] “script 432 can ... invoke compiler 414 with ... compiler optimization options enabled … to generate optimized object code 422”) to indicate the plurality of binary rewriter optimization combinations to the compiler (Ward par. [0019] “The option sets are chosen at control 13a … to optimize the different performance metrics”; AAPA par. bridging pp. 9-10 "Example optimization controls include constant propagation; code shrinking and specialization”). Those of ordinary skill in the art would have been

motivated to do so in order to automate what is disclosed as a manual process (i.e. Ward par. [0019] “The option sets are chosen at control 13a by the user using a graphical user interface 17”).

Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 2001/0032332 to Ward et al. (Ward) in view of US 2004/0117779 to Lagergren (Lagergren) in view of US 5,901,310 to Rahman et al. (Rahman).

Claim 20: The rejection of claim 13 is incorporated; further Ward discloses the code size includes at least one of code size (par. [0020] “”such as size”), compressed file size, and memory footprint.

Ward and Lagergren do not teach the binary comprises a firmware driver.

Rahman teaches that compressed file size is an important characteristic of a firmware driver (col. 1, lines 48-51 “virtually increases the size of the nonvolatile semiconductor memory …. by storing the firmware in compressed form”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to optimize a firmware driver for compressed size (Ward par. [0020] “”such as size”; Rahman col. 1, lines 48-51 “virtually increases the size of the nonvolatile semiconductor memory …. by storing the firmware in compressed form”) using Ward’s

system. Those of ordinary skill in the art would have been motivated to do so to minimize the amount of storage space, and thus the cost, required for the driver (Rahman col. 1, lines 48-51 “virtually increases the size of the nonvolatile semiconductor memory …. by storing the firmware in compressed form”; col. 1, lines 25-27 “Providing a larger ROM bears a greater cost”).

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JASON MITCHELL whose telephone number is

(571)272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/
Primary Examiner, Art Unit 2193